

AP Computer Science Principles

Mr. Kercheval, Room N-117

Contact Information:

email: adam.kercheval@jeffcoschools.us

Google Classroom code: hk5r2mk

Welcome to AP Computer Science Principles! Please read all the class expectations and if you have any questions, don't hesitate to ask.

Course Content:

We will be following the course developed by the University of California, Berkeley, called the [Beauty and Joy of Computing](#). Our work will be the same, though we will implement it in C++ instead of in Snap.

Required materials:

- Notebook paper and pen/pencil.
- Student chromebook or personal computer with a web browser. Please bring it with you to class every day.

Quizzes/Tests: This class will involve unit tests and unit projects, which are weighted the same. I will also give frequent, short quizzes to keep track of how you're learning content, and to prepare you for the types of questions you will see on the AP test. If you miss a quiz or a test, you must make it up within one week. Please contact me if you need an extension or exception to this rule.

Homework: Many homework assignments will consist of small, non-coding-based work to reinforce the day's topics, and will be due the following class period. Each week on block day I will also assign coding-based assignments for you to complete in class and over the weekend.

Projects: Each unit, we will complete a project of larger scale than the weekly homework assignments. I will often encourage you to complete these projects with a partner. **I anticipate that these projects will be challenging, and I will hold you to a high standard when assessing your submissions.**

Late Work Policy: Late homework will be accepted until the last day of the unit in which it is assigned, for 80% credit.

Grades: Grades are weighted and scaled based on the following breakdowns:

Weight:		Scale:	
Unit Projects/Tests:	60%	90 – 100%	A
Homework/Quizzes:	25%	80 – 89%	B
Final Exam:	15%	70 – 79%	C
		60 – 69%	D
		59% and below	F

Axioms/Advice: Many parts of computer science have a steep learning curve, and you may need to re-frame how and when you work academically in order to be successful in this class. My advice for computer science students is to:

- 1. Start early.** You will not be able to complete a project if you start it the night before it is due. Your software will be easier to understand, less error-prone, and more effective for you as a learner if you spread the work out. Spend the first day or so thinking about the project and working out procedures and algorithms in English or using physical objects before you attempt to program anything.
- 2. Write a little, test a little.** Separate your program into small, easily testable chunks that you can piece together. Finding a bug in 5 lines of code is much easier than finding a bug in 500.
- 3. Use resources and ask for help.** It's nearly impossible to write a program without external insight or support. Talk to others about bugs you encounter, ask Mr. Kercheval for help, ask Google and Stack Overflow for debugging tips (more details on what is and isn't allowed when discussing your code with others are in the course academic integrity agreement).

Academic Integrity:

Computer science is an interesting field because of the ease with which code can be copied from the internet, or, more recently, generated by artificial intelligence (AI) programs. In line with the College Board's outlook on AI, **artificial intelligence may be used** for code development and debugging in this course. Beware, however, that AI programs tend to be limited in the quality of code they produce. You are responsible for both citing and thoroughly understanding any software you submit that contains code that you did not write. More details on this policy are available at n117.dev/integrity_csp.pdf.